Change Detection in Unlabeled Optical Remote Sensing Data Using Siamese CNN

Rachid Hedjam, *Member, IEEE*, Abdelhamid Abdesselam, and Farid Melgani

***Abstract*—In this article, we propose a new semisupervised method to detect the changes occurring in a geographical area after a major damage. We detect the changes by processing a pair of optical remote sensing images. The proposed method adopts a patch-based approach, whereby we use a Siamese convolutional neural network (S-CNN), trained with augmented data, to compare successive pairs of patches obtained from the input images. The main contribution of this work lies in developing an S-CNN training phase without resorting to class labels that are actually not available from the input images. We train the S-CNN using genuine and impostor patch-pairs defined in a semisupervised way from the input images. We tested the proposed change detection model on four real datasets and compared its performance to those of two existing models. The obtained results were very promising.**

***Index Terms*—Remote sensing change detection (CD), semisupervised CD, Siamese convolutional neural network (CNN).**

I. Introduction

In REMOTE sensing, the main objective of change detection (CD) is to compare two or more images of the same geographical region acquired at different times to detect significant spectral differences between the pixels across the images [1]. Usually this can be solved either by a supervised or an unsupervised approach. A supervised CD requires a reference data to train and optimize the parameters of the model [2]. In contrast, an unsupervised CD is based on the difference image (DI) between the input images to generate the change map [3]. CD techniques can be also grouped into pixel-based, object-based, and data mining-based. A good review of these traditional techniques can be found in [1], [4], and [5]. Although these methods use the contextual spatial and spectral information of remote sensing images, the traditional manual design of image features is a tedious and complex task and it is mostly performed on the basis of domain-specific knowledge [6]–[9]. The explosive growth of available remote sensing data and the increased processing power afforded by graphical processing units has led to the rise of advanced techniques based on deep learning (DL) [10], [11]. These techniques are becoming increasingly important due to their effectiveness in automatic learning of discriminative features. Recently, a number of DL-based CD techniques have been proposed to solve CD problems, including fusion feature extraction and deep convolutional neural network (CNN) [12], deep belief network with fuzzy ontologies and multiscale analysis [13], recurrent neural networks and long short-term memory [14], dual-dense convolutional network [15], deep CNN for multimodal remote sensing images [16], semisupervised Siamese ANN [17], to name a few of them.

The proposed CD method aims to detect spatial changes that have occurred in a geographic region using two temporal images. The first image is acquired before the date of the event causing the change and the second is acquired after. This method adopts a patchwise matching between two images to find out whether the textural structure of the images at the region defined by the patches has undergone a change or not. It adopts the Siamese neural network to calculate the difference between the two images. Changed image regions are represented by larger values in the difference map compared to unchanged regions. The main difference with existing Siamese CNNs (S-CNNs) resides in the way we designed training data, i.e., genuine patch-pairs that represent the unchanged regions and impostor patch-pairs that represent the changed regions. During the training phase, only the first image is used. It undergoes several mathematical transformations then it is divided into patches to generate the training genuine patch-pairs. However, when the changed regions are relatively smaller than the unchanged regions, or are difficult to determine, it becomes hard to collect a sufficient number of training impostor patch-pairs, and therefore the most important requirement for CNN training (the need for large data) is not met. To deal with this problem, we propose substituting the second image with external images having a texture similar to the type of change to be treated. Basically, pairing patches from external images with patches from the first image makes it possible to define an alternative source for generating large training impostor patch-pairs. Overall, the aim is to build and train an S-CNN to learn, using instances of genuine and impostor patch-pairs, a similarity function, which takes two patches as input and expresses how similar they are [18]. In this work, the word "semisupervised" refers to the use of external images (from other image domains) to generate impostor patch-pairs. The proposed training scenario is inspired by some similar works on cross-domain adaptation and transfer learning that have been applied in many research works [19]–[21]. Therefore, our method requires nothing more than knowing the type of

Manuscript received March 31, 2020; revised June 9, 2020; accepted July 4, 2020. Date of publication July 14, 2020; date of current version August 3, 2020. *(Corresponding author: Rachid Hedjam.)*

Rachid Hedjam and Abdelhamid Abdesselam are with the Department of Computer Science, Sultan Qaboos University, Muscat 123, Oman (e-mail: rachid.hedjam@squ.edu.om; ahamid@squ.edu.om).

Farid Melgani is with the Department of Information Engineering and Computer Science, the University of Trento, I-38123 Trento, Italy (e-mail: melgani@disi.unitn.it).

Digital Object Identifier 10.1109/JSTARS.2020.3009116

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

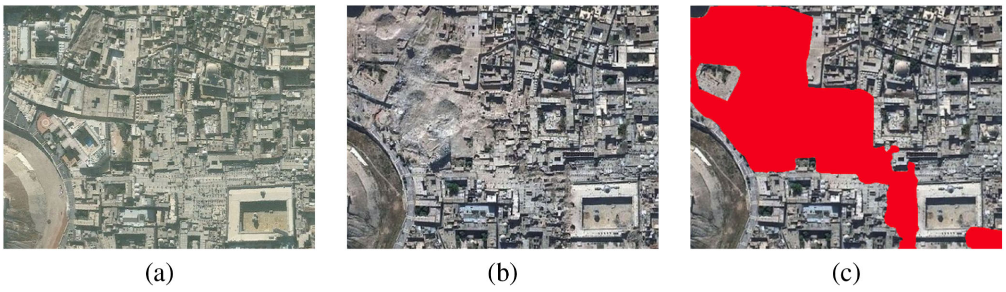


Fig. 1. Two satellite images. (a) Image before change. (b) Image after change. (c) Image (b) with changed areas masked in pink color. The mask is created manually based on the difference in perception between image (a) and image (b). Credit: DigitalGlobe.

change to be addressed, so that the substituting images will be selected accordingly. Moreover, we are addressing specific types of changes, those consisting of distortion of the image contents (geometrical deformation as shown in Fig. 1). In other terms, it focuses on the spatial difference between the input images and not on the spectral difference. It is therefore designed to process color or grayscale images. More details are given in Section IV.

The rest of this article consists of the following sections. Section II reviews related CD methods. Section III describes and motivates the problem addressed in this article. Section IV describes the proposed CD method. The experiments and results are reported and discussed in Section V. Finally, the conclusion is given in Section VI.

II. Related Works

Extracting relevant image features is necessary for obtaining accurate CD results. As mentioned above, traditional methods that are mainly based on hand-designed features have proven to be very limited especially in complex contexts. In recent years, a significant progress has been made in the field of computer vision, pattern recognition, and machine learning, eventually leading to the development of efficient automatic feature extraction techniques. DL is the most outstanding result of this progress and remote sensing is one of the areas that has benefited from it. CNNs are very powerful DL techniques capable of automatically extracting highly discriminating features [7], [22]. Thanks to this, numerous research works have given birth to powerful methods of CD in remote sensing. For instance, in [12], CNN features are extracted through different convolutional layers, normalized and concatenated in a single feature vector. Then, a change map is computed using a pixelwise Euclidean distance between the obtained feature vectors of the two input images, after reshaping them. The main advantage of [12] is that it extracts the features of each layer rather than waiting for the last layer to extract the global feature vector, in which the mid-level features are lost due to subsampling (pooling). The limitation of this method comes from the fact that a single pretrained CNN cannot extract features that are relevant to all types of changes. In [23], a symmetrical CNN is proposed to detect changes from two heterogeneous images (optical and radar). Each image is fed to one side of the network, to be transformed into a more coherent feature vector. The resulting vectors are used to calculate the DI and generate the binary change map. The parameters of the entire network are learned by optimizing a coupling function. Gong *et al.* [24] proposed an unsupervised CD in multispectral imagery based on generative adversarial networks (GANs). The method first calculates the DI using change vector analysis and an intermediate binary change map and uses them to train and use two adversarial networks (GAN) to generate a better DI. The latter is thresholded by an unsupervised clustering algorithm to generate the final change map. Zhang *et al.* [25] proposed an S-CNN to detect candidate buildings and trees changes between different epochs using 3-D laser and 2-D aerial images of the same area. The method converts the two images into gray images, splits them into patches, then feeds them into an S-CNN along with labels. The trained S-CNN is finally used to detect candidate changes between the two epochs. In [26], the optimal representations of changed areas are learned using an unsupervised stacked autoencoder, followed by a label aggregation performed in the feature space before classification. Although the method has proven to be effective with only a small number of labels, but it still needs to be supervised. In [27], a fully atrous CNN is proposed and used to learn the classification of land cover, in which robust features of very high resolution images are extracted by an encoder. In a second step, the change map is calculated by pixelwise distance between the features extracted from the input images. In [28], Chen *et al.* combined conditional GANs [29] with U-Net [30] to solve the problem of dense CD in remote sensing without relying on preprocessing or postprocessing steps. The authors argued that by feeding a fully connected network with whole images, instead of patch by patch, the method improves the detection performance.

III. Problem Statement

Multitemporal optical remote sensing images show different types of minor and major changes. Minor changes can be caused by different factors such as imperfection in the acquisition systems, difference in acquisition platforms, weather disturbance, etc. This means that in general, there is always a small difference between any two consecutive images (of the same area), especially if the temporal resolution is small. Major changes are mainly due to serious effects such as human-made construction and destruction, natural disasters such as floods,

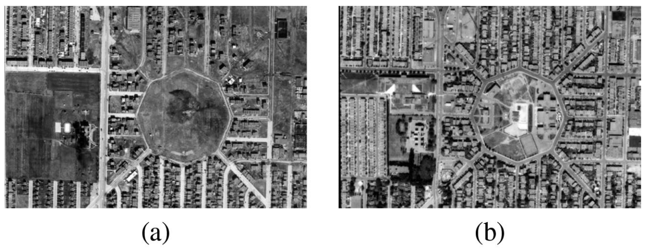


Fig. 2. Historical aerial image. City of Montreal (Canada) in the 1950s. (a) Before change made by construction. (b) After change. Photographs provided by Archive of Montreal.

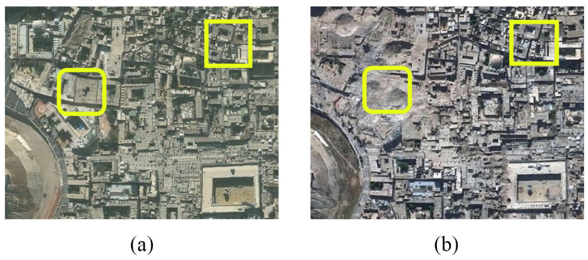


Fig. 3. Patchwise pairing. (a) First image. (b) Second image. Sharp patches (genuine patch-pair) represent unchanged areas, whereas rounded patches (impostor patch-pair) represent changed areas.

earthquakes, fires, to name a few. In Fig. 1(c), a mask is created manually based on a visual inspection of the major changes between Fig. 1(a) and (b). The masked area represents the change caused by a military aircraft bombing. The images show part of Aleppo city in Syria taken, respectively, on November 21, 2010 and October 22, 2014. Minor changes can, in turn, be noticed by visual inspection of the images 1(a) and (b). In this research work, we specifically address this type of problems. The images used to conduct our experiments consist of grayscale and color images captured by recent satellites, and historical aerial photographs taken by airborne cameras. An example is shown in Fig. 2. For this kind of images, relying on pixelwise DI to generate the change map is not a suitable approach, because it is difficult to distinguish between the changed and unchanged areas based solely on pixel intensity. As a remedy, we propose analyzing the geometric deformation of the image contents. Edge orientation and texture can be promising features. While computer vision and image processing have provided us with many texture analysis techniques in recent years, DL, specifically CNN, has been proven to be a very powerful tool for feature extraction from complex textured images [31].

IV. Proposed Method

Let I and J be the first and second images, respectively. Let us define by "patch" a square region in an image, and by a "patch-pair" a twin of patches representing the same area in both images as illustrated in Fig. 3. Patch-pairs that represent minor changes are called genuine pairs, whereas patch-pairs that represent major changes are called impostor pairs. In this work, the CD problem is solved as follows. 1) Calculate the local

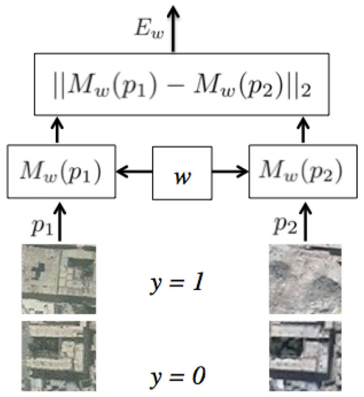


Fig. 4. CNN-Siamese neural network model. (p_1, p_2) stand for a patch-pair. $y = 0$ means that the patch-pair is genuine, whereas $y = 1$ means that the patch-pair is impostor.

similarities (or differences) between the images I and J. This is done by calculating the distance between each patch in I and its twin in J. At the end of this process, the DI is produced. 2) Segment DI into two classes. Optimistically, the class with the lowest similarity values (or highest difference values) represents the changed areas and the class with the highest similarity values represent the unchanged areas. However, the quality of DI depends mainly on two factors: 1) the relevance of the features extracted from the patches and 2) the optimization of the distance used to calculate the similarity between features. Therefore, the goal is to design a whole system that automatically extracts robust features with higher discrimination power and also optimizes the distance between these features. S-CNN [32] is the outstanding approach used to meet these two requirements. S-CNN consists of predicting whether the input patch-pairs are genuine or impostor.

Our proposed method consists of a training followed by a testing phase. In the training phase, we establish a trainable nonlinear function that maps the patch-pairs to points in a low dimensional feature space so that the distance (e.g., Euclidean distance) between these points is small if the twin patches are genuine and large if they are impostor. The automatic learning process of the distance is achieved by training two identical CNNs sharing the same set of weights. This is called Siamese network [32]. In the test phase, only a CNN of one side is used to detect the change between I and J.

A. Siamese Network Model

In the proposed method, an S-CNN consists of two identical copies of subnetworks sharing the same weights w, and a top network that computes the patch-pair similarity. An illustration of an S-CNN is shown in Fig. 4. Each subnetwork is represented by a CNN that generates a feature vector describing the properties of the corresponding input patch. The obtained two feature vectors are compared using the Euclidean distance (other distances can be used as well) to produce the final output of the network. Formally, let (p_1, p_2) be an image patch-pair used to train the network, and y be a binary label of the pair, such that

$y = 0$ if (p_1, p_2) is genuine (noted $(p_1, p_2)^+$), and $y = 1$ if it is impostor (noted $(p_1, p_2)^-$). Let w be the shared weight vector to be learned, and let $M_w(p_1)$ and $M_w(p_2)$ be the outputs of a nonlinear mapping function for p_1 and p_2, respectively. The top network aims to compute the distance (i.e., similarity) between p_1 and p_2 through an energy function E_w as follows:

$$E_w(p_1, p_2) = ||M_w(p_1) - M_w(p_2)||_2. \tag{1}$$

B. Learning Approach

Given a training set of N genuine and impostor patch-pairs $\{(p_1, p_2)_i; 1 \leq i \leq N\}$ along with their labels $Y = \{y_i \in [0, 1]\}$. We propagate the patch-pairs with their labels through the Siamese network to learn their feature vectors and then compute the loss function (error or objective function) $\mathcal{L}$ defined as follows:

$$\mathcal{L}_w(\{(p_1, p_2)_i\}, Y) = \sum_{i=1}^{N} \left[(1 - y_i)\mathcal{S}_w(p_1, p_2)_i^+ + y_i \mathcal{D}_w(p_1, p_2)_i^-\right]. \tag{2}$$

The loss function in (2) consists of two terms, one that acts on the genuine pairs, and another that acts on the impostor pairs. The first term (noted $\mathcal{S}_w$) penalizes large distances within the impostor pairs and consequently promotes the genuine pairs through the following formula:

$$\mathcal{S}_w(p_1, p_2)_i^+ = E_w(p_1, p_2)_i^+; \text{ if } y_i = 0 \tag{3}$$

where (i) indicates the ith sample. The second term (noted $\mathcal{D}_w$) penalizes small distances within the genuine pairs and consequently promotes the impostor pairs through the following formula:

$$\mathcal{D}_w(p_1, p_2)_i^- = e^{-E_w(p_1, p_2)_i^-}; \text{ if } y_i = 1. \tag{4}$$

The two terms $\mathcal{S}_w$ and $\mathcal{D}_w$ are competing in a manner that the minimization of total energy $\mathcal{L}$ leads to decrease energy of genuine pairs, and increase energy of impostor pairs. The loss function is derived with respect to the shared weight vector w, using standard back-propagation algorithm. The parameter vector is updated with a stochastic gradient method using the sum of the gradients contributed by the two subnetworks [33].

C. Approach for Generating the Training Dataset

As stated in Section III, due to the lack of changed and unchanged class labels, generating training dataset is the most critical task in our work. In the ideal case, we assume that a set of multitemporal images is available and it is divided into two subsets, $\mathcal{I} = \{I_i\}_{i=1}^{t}$ (acquired before the change date) and $\mathcal{J} = \{J_j\}_{j=t+1}^{t+n}$ (acquired after). The subscript stands for the image acquisition time. In case where some changed areas are well predefined (e.g., by a mask of labels), the training patch-pairs can be sampled with their labels by simultaneously patch-sliding two images (I_i, J_j) from left-top to bottom-right. This process is repeated on other pairs of images to have a more representative training data. Unfortunately, when the changed regions are not defined, which is our case, the unsupervised process is not guided by a mask of labels, and therefore cannot blindly guarantee that the impostor patch-pairs are sampled from the changed regions, and the genuine patch-pairs are sampled from the unchanged regions. To work around this problem, we adopt two different strategies, one for each patch-pair type. The first strategy that concerns the genuine patch-pairs, comes from the fact that any J_j is somewhat a "temporal continuation" of any I_i (i.e., time series). Therefore, they are similar inside unchanged regions (as defined by genuine patches in Fig. 3). In this case, $\mathcal{I}$ will suffice to sample genuine-pairs without having to use $\mathcal{J}$. Formally, $(p_1, p_2)^+ \in (I_i, I_k), \forall i, k \in [1, \ldots, t]$. Generating genuine patch-pairs is therefore a straightforward unsupervised task. In contrast, generating impostor patch-pairs requires that the two patches p_1 and p_2 should be generated from a region affected by the change, which is unfortunately unknown for us. To go around this problem, we propose not to use the images $\mathcal{J}$ at all. Instead, we suggest to sample the patches p_2 from other external images with textures that resemble to that regions affected by the damage would have. It is worth to note that before creating the patches, we first visually compare, using the Gimp Graphical tool, the resolution of external images to that of the input images, and downscale or upscale the external images if the difference is large. It is therefore preferable to use external images with a resolution approximately similar to that of the input images.

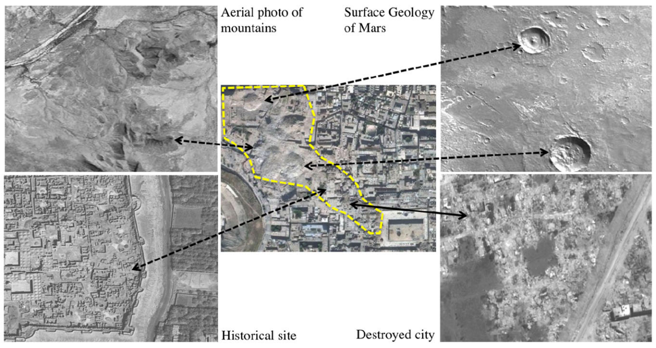
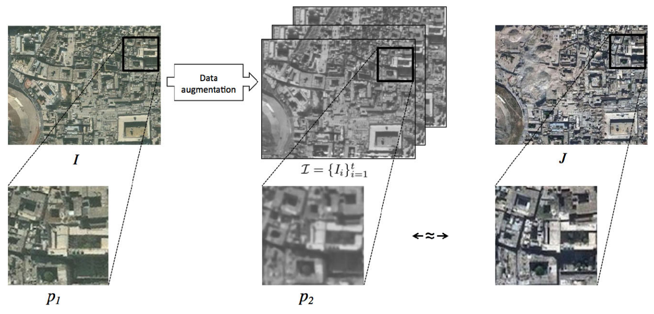


Fig. 5. Example of some external textured images that can be used to generate the impostor patch-pairs. The image in the middle is the image after the change shown in Fig. 3(b). The yellow dotted curve is included to delimit approximately the area affected by the major change. The arrows are included to show the similarity between the texture of the changed regions and that of the external images.

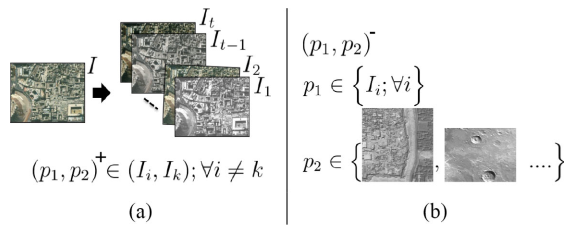
Practically we crop as much as we can p_2 patches from the external images. Some examples of these images are shown in Fig. 5.

Actually, we do not have two subsets of multitemporal images; we only have two images noted I and J. And as explained at the beginning of this section, the image J will not be considered in the training phase. So how will it be possible to design training genuine-pairs using I? In order to generate the training genuine patch-pairs, we need other images to be coupled to the image I. Images that deviates from it with minor changes. To do this, we propose to artificially create $\mathcal{I} = \{I_i\}_{i=1}^{t}$ and generate the training genuine patch-pairs in the same way as described above. The simplest idea to create $\mathcal{I}$ is to apply several small



Data augmentation
I
$\mathcal{I}=\{I_i\}_{i=1}^{t}$
J
p_1
p_2
$\leftarrow\approx\rightarrow$

Fig. 6. Data augmentation. The images $\mathcal{I}$ are created from the image I. (p_1, p_2) is a genuine patch-pair. They are similar to the image J within the unchanged regions.



I
I_t
I_{t-1}
I_2
I_1
$(p_1,p_2)^{+} \in (I_i, I_k); \forall i \neq k$
(a)
$(p_1,p_2)^{-}$
$p_1 \in \{I_i; \forall i\}$
$p_2 \in \{ \ldots \}$
(b)

Fig. 7. Illustrative example of the training data generation process. (a) Genuine patch-pairs. The images $\{I_1, \ldots, I_t\}$ are the transformations of the image I. The whole set is used to generate the genuine patch-pairs. (b) Impostor patch-pairs. The patches p_2 are extracted from the external images, whereas p_1 is extracted from the image I_i, then coupled to form the impostor patch-pairs.

Algorithm 1: Proposed Change Detection Method.

```
    Input 1: input images, $I, J$.
    Input 2: $\{E\}$: predefined external patches.
    Output: binary change map, $M$.
 1: procedure
 2:   # generate genuine patch-pairs.
 3:   Generate $\mathcal{I}$ from $I$.
 4:   $\mathcal{I} \leftarrow I$, #include $I$ in $\mathcal{I}$.
 5:   Extract $\{(p_1,p_2)^+\}$ from
      $\{(I_i, I_k)\}, \forall I_i, I_k \in \mathcal{I}$.
 6:   # generate impostor patch-pairs.
 7:   Extract $\{(p_1,p_2)^-\}$. $p_1 \in \mathcal{I}$, $p_2 \in E$.
 8:   # Train the Siamese CNN ( S-CNN ).
 9:   # Detect change using $I$ and $J$.
10:   for each $(p_1,p_2)_a$ in $(I, J)$ do
11:   $f_1 \leftarrow \mathtt{CNN}(p_1)$ #extract features of $p_1$ by
      CNN.
12:   $f_2 \leftarrow \mathtt{CNN}(p_2)$ #extract features of $p_2$ by
      CNN.
13:   $D(a) \leftarrow \|f1 - f2\|_2$ #Euclidean distance.
14:   # Reshape $D$ to have same size as $I$
      and $J$.
15:   # Segment $D$ using Otsu's algorithm.
16:   $M \leftarrow \mathtt{Otsu}(D)$
```

mathematical transformations to the image I, such as rotation, scaling, offset, stretching, blurring, noise, etc. In other terms, we perform data augmentation on I. By artificially creating $\mathcal{I}$, we solve two problems at once. 1) Generated images will most probably resemble to image J inside the unchanged regions. This is illustrated in Fig. 6. It turns out that the patch p_2, a transformed version of the patch p_1, somehow resembles its twin patch belonging to the image J. The simulated images $\mathcal{I}$ and the image I are therefore used to generate the genuine patch-pairs as described in the first paragraph of this section. Fig. 7 illustrates the entire training data generation process. 2) By creating $\mathcal{I}$, we increase the size of the data (i.e., data augmentation), which is in fact a recommendation for the training of any DL method. An illustration of the whole training data generation process is given in Fig. 8(a).

D. Test Phase

Because the two sub-CNNs will be identical after being trained [34], when we test our model, only one of the two sub-CNNs is evaluated. The output of this is the feature vector for the input patch. More specifically, the test process involves the following tasks. 1) Take a patch from the first image and another patch from the same position in the second image, feed them to the sub-CNN to obtain their feature vectors, then normalize them in the range [0, 1]. 2) Calculate the Euclidean distance between them. We repeat the same process for all the image patch-pairs results in the so-called DI. 3) Classify or segment the pixels of DI into two classes (using, for example, Otsu's algorithm). The pixels having the highest values belong to changed areas and the pixels having the lowest values belong to unchanged areas, as illustrated in Fig. 8(b).

Below is the overall algorithm of the proposed CD model. An illustration of the training and test phases of the proposed S-CNN CD model is also given in Fig. 8. As shown, the proposed model requires as input the image acquired before the change (i.e., I) and the database of patches extracted from the external images with a texture similar to the change in question. However, during the test phase, both images I and J, acquired before and after the change, respectively, are used.

V. Experimental Results

A. Dataset Description

Four datasets are used to validate the proposed CD method. The first dataset[1] is a bitemporal image showing part of Aleppo city in Syria before and after a military aircraft bombing (see Fig. 9). The two images are taken, respectively, on November 21, 2010 and October 22, 2014. The second dataset[1] is a cropped bitemporal image showing a part of the entire district of Masaa Al Arbaeen in the city of Hama (Syria), which was flattened after a military strike between September 27, 2012 and October 23, 2012. A total of 3256 buildings were reduced to rubble (see Fig. 10). The third dataset[2] consists of before and after satellite images of the Syrian nuclear reactor at Al-Kibar, which was reportedly struck by Israel in 2007. The two images are taken

[1]Online. [Available]:https://www.bbc.com/news/world-middle-east-31871568

[2]Online. [Available]: https://www.aljazeera.com/indepth/features/2013/05/20135512739431489.html

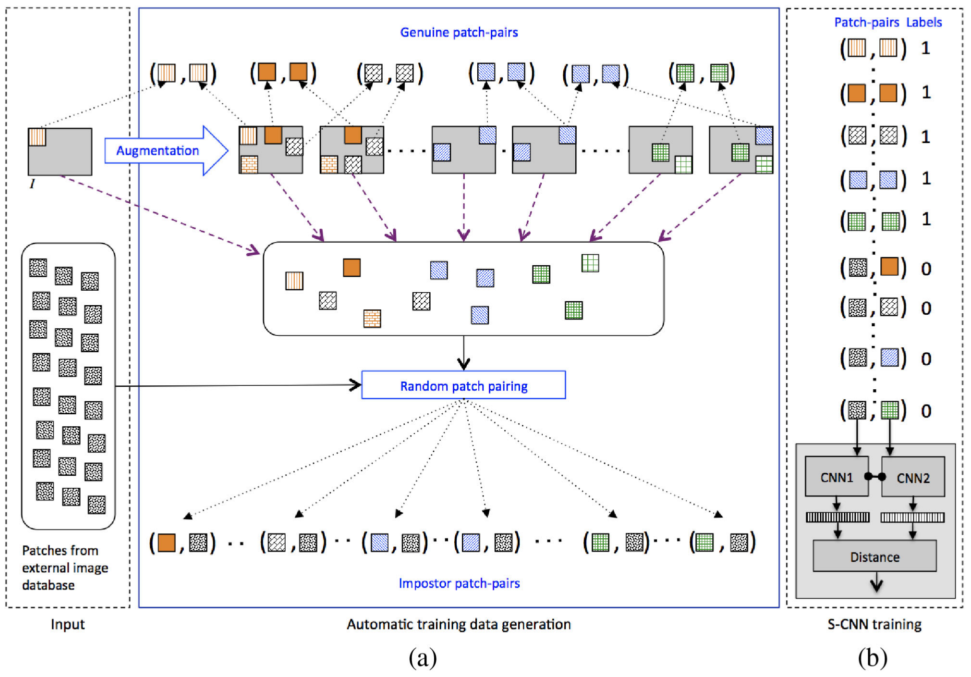


Fig. 8. Illustration of the proposed S-CNN CD process. (a) Training phase. The black dotted arrows mean the generation of patch-pairs. (b) Test phase. Since CNN1 and CNN2 are identical after the training phase, one of them is used in the test phase.

on October 18, 2007 and October 24, 2007 (see Fig. 11). The fourth dataset[3] consists of two cropped images of Montreal city acquired in the 1950s (see Fig. 12).

For objective evaluation purposes, reference binary images are created manually based on the visual comparison of the twin images. Areas with major changes are labeled with white pixels, whereas others are labeled with black pixels. The reference images are also shown in Figs. 9–12. They are created manually using Gimp Graphics editor.[4] The datasets along with their reference images can be found on GitHub.[5]

B. Model Parameters and Setting

We have designed, trained, and tested several S-CNN architectures. The one that led to the best performance is described as follows: The CNN in each side of the S-CNN is composed of two convolutional layers. Each layer performs ten 3×3 convolutions without padding. The activation function ReLU is used and followed by a pooling layer that performs 2×2 max pooling. The last layer is flattened into a layer of 1960 features (real values) used as the feature vector of the input patch. Training the proposed S-CNN involves the following steps. Step 1) Case 1) If the two input images are in color, we use them as they are. Case 2) If they are both grayscale, we convert them to three-channel images, where each channel holds the original grayscale image. Case 3) If one image is color and the other is grayscale, we convert the color image to a grayscale image and then convert the two grayscale images to three-channel images as explained in case 2). Step 2) We performed data augmentation for the first image (I) by applying, 1° left/right rotation, 2 pixels scaling (zoom-in and zoom-out), 3 pixels left/right/up/down shifting. In total 16 (i.e., $2 \times 2 \times 4$) images are created from I. Therefore, the image I, the 16 created images, and the external images shown in Fig. 5 are used to generate the training patch-pairs as described in Section IV-C. Step 3) We set the size of the patches to 64×64 pixels. By consequence, the total number of patch pairs used to train S-CNN depends on the size of the input images. Step 4) We split the composed dataset into 70% for training and 30% for validation.

During the test phase and in order to speed up the CD process, the sliding step is fixed at 5 pixels. Because of the patch-overlap caused, the DI produced will have a smaller size than that of the input images. Therefore, it will be resized (cubic interpolation is used) to the original size, before performing segmentation (i.e., thresholding).

C. Objective and Subjective Performance Evaluation

For objective evaluation purpose the proposed method is compared to two other models, the baseline image differencing (IDif) [35], and CNN hyper features-based model [12]. The outputs of each CD model are evaluated against the corresponding reference images based on true positive (TP), true negative (TN), false positive (FP), false negative (FN), Accuracy (Acc.) Recall (Rec.), Precision (Prec.), and F-score ($F1$). All of them are described in the Sikit-learn Python library.[6] The objective results of different methods are reported in Table I. we can see

[3]Provided by Open Data, Montreal.
[4]Online. [Available]: https://www.gimp.org/
[5]Online. [Available]: https://github.com/rh-gt/RS-data
[6]Online. [Available]: https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

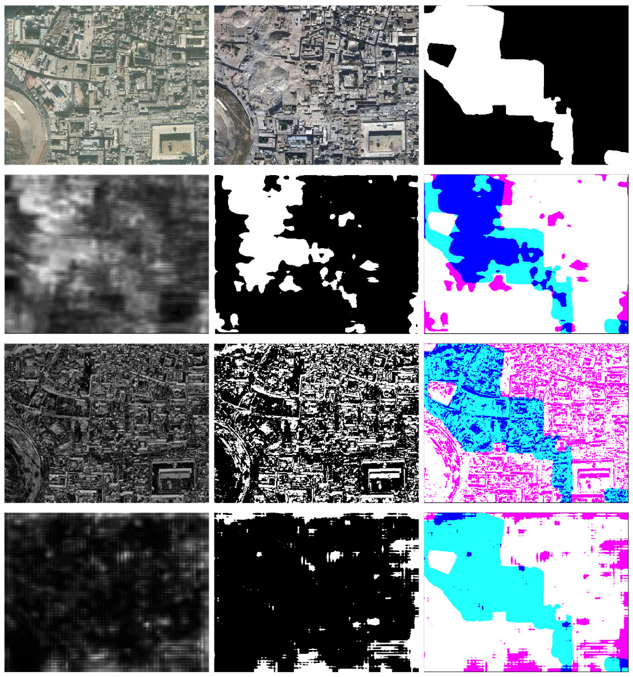


Fig. 9. Subjective comparison-Aleppo dataset. First row: Two input images and ground-truth. Second row: Proposed method's output. Third row: IDif model's output. Fourth row: [12] model's output. Left: DI (gray-scale). Middle: Binary change map (black/white). Right: Confusion map between the binary map and the ground-truth: (blue: TP; white: TN; magenta: FP; cyan: FN).

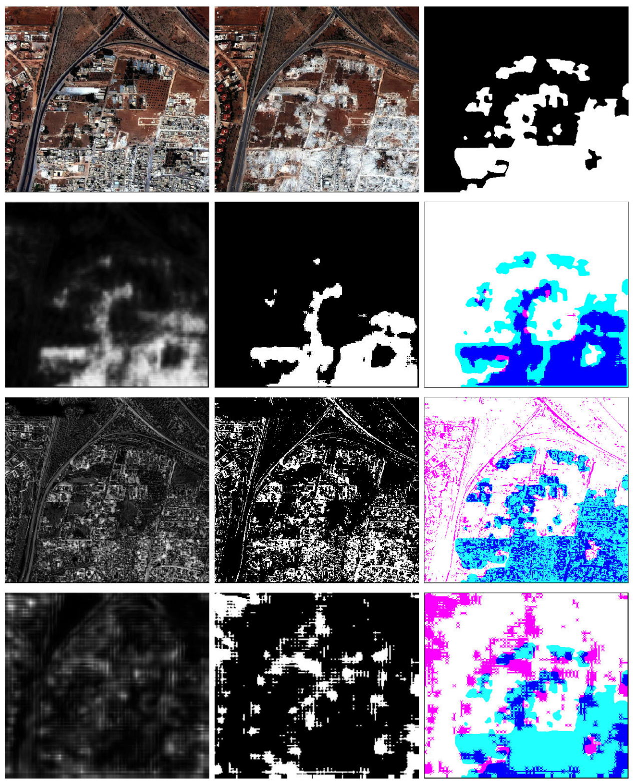


Fig. 10. Subjective comparison-Hama dataset. First row: Two input images and ground-truth. Second row: Proposed method's output. Third: IDif model's output. Fourth row: [12] model's output. Left: DI (gray-scale). Middle: Binary change map (black/white). Right: Confusion map between the binary map and the ground-truth: (blue: TP; white: TN; magenta: FP; cyan: FN).

TABLE I
PERFORMANCE OF DIFFERENT CD METHODS, IDIF, [12], AND PROPOSED, BASED ON TP, TN, FP, FN, ACC., REC., PREC., AND F-SCORE ($F1$)

Meth.	TP	TN	FP	FN	Acc.	Rec.	Prec.	F1
	Aleppo							
IDif	18704	72998	41789	36497	0.54	0.34	0.31	0.32
[12]	2684	97686	17101	52517	0.59	0.05	0.14	0.07
Ours	**32727**	**103268**	**11519**	**22474**	**0.80**	**0.59**	**0.74**	**0.66**
	Hama							
IDif	27110	122439	15755	40804	0.73	0.40	0.63	0.49
[12]	14413	115139	23055	53501	0.63	0.21	0.38	0.27
Ours	**36451**	**137110**	**1084**	**31463**	**0.84**	**0.54**	**0.97**	**0.69**
	Al-Kibar							
IDif	**3057**	45863	13563	**3053**	0.75	**0.50**	0.18	0.27
[12]	719	**58088**	**1338**	5391	**0.90**	0.12	**0.35**	0.18
Ours	2743	54335	5091	3367	0.87	0.45	**0.35**	**0.39**
	Montreal							
IDif	23374	52803	12433	64990	0.50	0.26	0.65	0.38
[12]	6041	56216	9020	82323	0.41	0.07	0.40	0.12
Ours	**35664**	**59025**	**6211**	**52700**	**0.62**	**0.40**	**0.85**	**0.55**

that the proposed method outperforms others in all experiments based on all the performance measures. Hama and Aleppo are the easiest datasets to process, and Montreal and Al-Kibar are the hardest. For example, one of the reasons why the Al-Kibar dataset is a tough case is that the texture and structure of much of the destroyed site are almost similar to that of the background of the second image (see Fig. 11). The results show that TN is higher than TP in almost all cases for all the methods. The reason is that the unchanged areas are relatively larger than the changed areas in almost all datasets. We can notice that the proposed method is better than others in terms of TP in almost all the cases, except in Al-Kibar dataset where IDif method is slightly better. Besides, the proposed method has the lowest FP in almost all the cases. This means that it causes fewer false alarms (parts shown with magenta in Figs. 9–12, third columns). This is also indicated by higher Prec. measures in all cases except Al-Kibar dataset, which is a challenge for all the methods. IDif is the method most causing false alarms, whereas [12] is the method that least detects real (positive) change. For example, compared to the other methods, the proposed method is able to detect the most of the parts of the changed areas in Hama and Aleppo datasets. This is well reflected by the values of Rec. measures. IDif method is better than the proposed method in terms of Rec. on Al-Kibar dataset but much worse in terms of Prec. due to the higher number of false alarms it generates. Overall, the proposed method is better than all in terms of $F1$ score, which is in fact a compromise between the miss-detection of real changes and false alarms.

For visual comparison purpose, the confusion maps of different methods are shown in Figs. 9–12, third columns. The confusion maps indicate that the proposed method has fewer magenta and cyan pixels compared to the others, which means less FP (false alarms) and FN (unable to detect change) values. The IDif method has more magenta pixels, i.e., it causes large false alarms, whereas [12] has more cyan pixels, i.e., it is less able to detect change.

In addition, we have studied experimentally the effect of patch size on the performance of the proposed method. We have

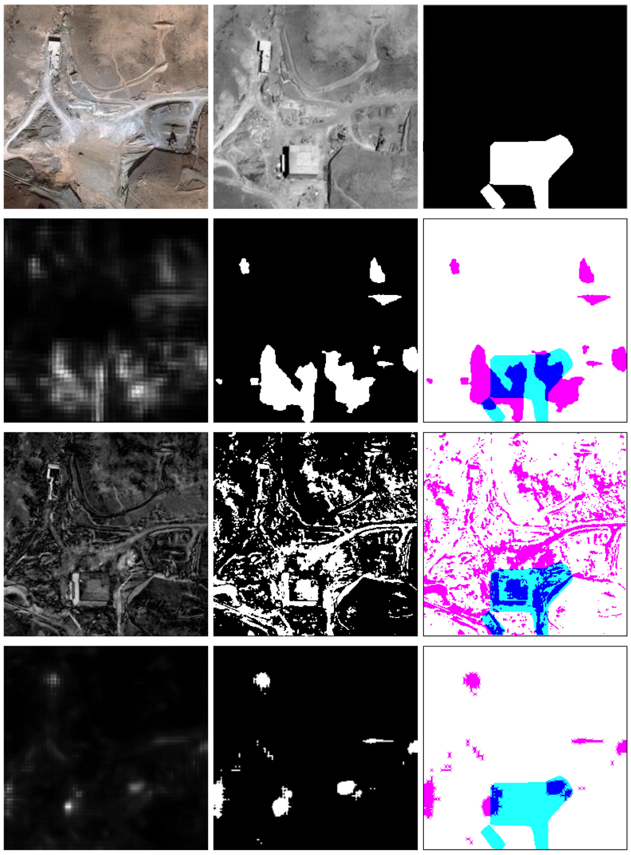


Fig. 11. Subjective comparison-Al-Kibar dataset. First row: Two input images and ground-truth. Second row: Proposed method's output. Third: IDif model's output. Fourth row: [12] model's output. Left: DI (gray-scale). Middle: Binary change map (black/white). Right: Confusion map between the binary map and the ground-truth: (blue: TP; white: TN; magenta: FP; cyan: FN).

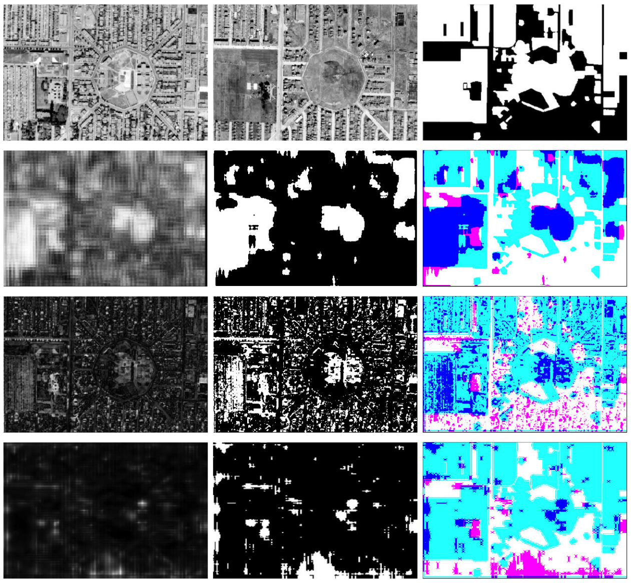


Fig. 12. Subjective comparison-Montreal dataset. First row: Two input images and ground-truth. Second row: Proposed method's output. Third: IDif model's output. Fourth row: [12] model's output. Left: DI (gray-scale). Middle: Binary change map (black/white). Right: Confusion map between the binary map and the ground-truth: (blue: TP; white: TN; magenta: FP; cyan: FN).

TABLE II
EFFECT OF PATCH SIZE ON THE PERFORMANCE OF THE PROPOSED CD MODEL

Patch size	TP	TN	FP	FN	Acc.	Rec.	Prec.	F1
32	33193	**59591**	**5645**	55171	0.60	0.38	**0.85**	0.52
64	**35664**	59025	6211	**52700**	**0.62**	**0.40**	**0.85**	**0.55**
128	32179	55416	9820	56185	0.57	0.36	0.77	0.49

Montreal dataset is used as an example.

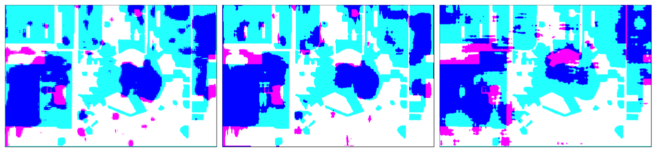


Fig. 13. Effect of patch size on the proposed CD model's output. Form let to right: 32×32, 64×64, 128×128 pixels (blue: TP; white: TN; magenta: FP; cyan: FN).

TABLE III
EFFECT OF THE TRAINING DATASET SIZE (EXPRESSED AS A PERCENTAGE OF THE TOTAL NUMBER OF GENERATED PATCH-PAIRS) ON THE PERFORMANCE OF THE PROPOSED MODEL

TrainSet size	TP	TN	FP	FN	Acc.	Rec.	Prec.	F1
80%	53003	101066	37128	14911	0.75	0.78	0.59	0.67
60%	44055	91626	46568	23859	0.66	0.65	0.49	0.56
40%	41181	94226	43968	26733	0.66	0.61	0.48	0.54

Hama dataset as an example.

conducted experiments with 32×32, 64×64, and 128×128 patches. We found that patches with a size of 64×64 pixels led to the best performance (see Table II). An example of the Montreal dataset is shown in Fig. 13. Same results were concluded for other datasets. We have found that the larger the patch size, the more false alarms (FP) are reported. Also, smaller FN is given by 64×64 patch-size. One of the reasons that the 64×64 patches work better, at least for the datasets used in this work, is that smaller patches (32×32) may not contain enough information so that the S-CNN can extract relevant features, whereas larger patches (128×128) may encompass parts of other objects leading to the introduction of false match between patches of input images. Setting the patch size requires prior information about the data to process.

We have also investigated the effect of the size of the training dataset on the performance of the proposed S-CNN. We conducted an experiment in which we trained and tested the model by varying the size of the training dataset. Table III lists the performance of the proposed model on the Hama dataset. Overall, larger training sets lead to better performance (at least for F1-score). This is almost always valid given that CNN-based models such as Siamese ANN require a large amount of data to be well trained [36].

VI. CONCLUSION

In this article, we proposed a new CD model that works on a pair of unlabeled optical remotely sensing images using an S-CNN. The challenge resulting from the lack of class labels is addressed by introducing a new method to generate appropriate training data needed by S-CNN, i.e., genuine and impostor patch-pairs. In a semisupervised way, the genuine patch-pairs are directly generated from the transformed maps of the first image (taken before the change), whereas the impostor patch-pairs are generated by pairing the first image with other external images having textures that resemble the change to be addressed. Therefore, our method requires nothing more than knowing the type of change to be addressed, so that the external images will be selected accordingly (found form the web, for example). We have analyzed subjectively and objectively the performance of the proposed method against two existing CD methods based on four real datasets and eight well-known performance scores. The results show that the proposed method is performing better. The advantage of the proposed CD method is that it operates automatically without resorting to class labels. However, this requires *a priori* knowledge of the type of change to be analyzed. It is actually a requirement to use the appropriate external images to generate the impostor patch-pairs. Finally, we would like to mention that the proposed method is designed to deal with only one type of change at a time. However, in order to adapt it to deal with several types of changes, we can train the S-CNN with external images with different textures that resemble the types of changes to be addressed. This will be the subject of future work.

Acknowledgment

The authors would like to thank the Archives of Quebec for providing historical aerial images of the city of Montreal.

References

[1] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin, "Review ArticleDigital change detection methods in ecosystem monitoring: A review," *Int. J. Remote Sens.*, vol. 25, no. 9, pp. 1565–1596, 2004.

[2] B. Demir, F. Bovolo, and L. Bruzzone, "Classification of time series of multispectral images with limited training data," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3219–3233, Aug. 2013.

[3] F. Bovolo, S. Marchesi, and L. Bruzzone, "A framework for automatic and unsupervised detection of multiple changes in multitemporal images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2196–2212, Jun. 2012.

[4] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *ISPRS J. Photogrammetry Remote Sens.*, vol. 80, pp. 91–106, 2013.

[5] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.

[6] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 4063–4067.

[7] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[8] Z. Pan, J. Xu, Y. Guo, Y. Hu, and G. Wang, "Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net," *Remote Sens.*, vol. 12, no. 10, pp. 1574–1590, 2020.

[9] M. Wang, K. Tan, X. Jia, X. Wang, and Y. Chen, "A deep Siamese network with hybrid convolutional feature extraction module for change detection based on multi-sensor remote sensing images," *Remote Sens.*, vol. 12, no. 2, pp. 205–222, 2020.

[10] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community," *J. Appl. Remote Sens.*, vol. 11, no. 4, 2017, Art. no. 042609.

[11] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 152, pp. 166–177, 2019.

[12] A. M. El Amin, Q. Liu, and Y. Wang, "Convolutional neural network features based change detection in satellite images," in *Proc. 1st Int. Workshop Pattern Recognit.*, 2016, vol. 10011, Art. no. 100110W.

[13] A. Argyridis and D. P. Argialas, "Building change detection through multi-scale GEOBIA approach by integrating deep belief networks with fuzzy ontologies," *Int. J. Image Data Fusion*, vol. 7, no. 2, pp. 148–171, 2016.

[14] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, pp. 506–527, 2016.

[15] W. Wiratama, J. Lee, S. Park, and D. Sim, "Dual-dense convolution network for change detection of high-resolution panchromatic imagery," *Appl. Sci.*, vol. 8, no. 10, pp. 1785–1797, 2018.

[16] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep Siamese convolutional network for optical aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[17] R. Hedjam, A. Abdesselam, and F. Melgani, "Change detection from unlabeled remote sensing images using Siamese ANN," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 1530–1533.

[18] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 4353–4361.

[19] M. M. Al Rahhal, Y. Bazi, T. Abdullah, M. L. Mekhalfi, H. AlHichri, and M. Zuair, "Learning a multi-branch neural network from multiple sources for knowledge adaptation in remote sensing imagery," *Remote Sens.*, vol. 10, no. 12, pp. 1890–1907, 2018.

[20] H. Lee, S. Eum, and H. Kwon, "Cross-domain CNN for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 3627–3630.

[21] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[22] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361. Cambridge, MA, USA: MIT Press, 1995.

[23] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, Mar. 2018.

[24] M. Gong, X. Niu, P. Zhang, and Z. Li, "Generative adversarial networks for change detection in multispectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2310–2314, Dec. 2017.

[25] Z. Zhang, G. Vosselman, M. Gerke, D. Tuia, and M. Y. Yang, "Change detection between multimodal remote sensing data using Siamese CNN," 2018, *arXiv:1807.09562*.

[26] S. De, D. Pirrone, F. Bovolo, L. Bruzzone, and A. Bhattacharya, "A novel change detection framework based on deep learning for the analysis of multi-temporal polarimetric SAR images," in *Proc. Int. Geosci. Remote Sens. Symp.*, Jul. 2017, pp. 5193–5196.

[27] C. Zhang, S. Wei, S. Ji, and M. Lu, "Detecting large-scale urban land cover changes from very high resolution remote sensing images using CNN-based classification," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 4, pp. 189–204, 2019.

[28] Y. Chen, X. Ouyang, and G. Agam, "ChangeNet: Learning to detect changes in satellite images," in *Proc. 3rd ACM SIGSPATIAL Int. Workshop AI Geographic Knowl. Discovery*, New York, NY, USA, 2019, pp. 24–31.

[29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1125–1134.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[31] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From bow to CNN: Two decades of texture representation for texture classification," *Int. J. Comput. Vision*, vol. 127, no. 1, pp. 74–109, 2019.

[32] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 737–744.

[33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2005, vol. 1, pp. 539–546.

[34] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[35] S. Minu and A. Shetty, "A comparative study of image change detection algorithms in MATLAB," *Aquatic Proc.*, vol. 4, pp. 1366–1373, 2015.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

Rachid Hedjam (Member, IEEE) received the M.Sc. degree in computer science from the University of Montreal, Montreal, QC, Canada, in 2009, and the Ph.D. degree in computer science from the University of Quebec, Trois-Rivieres, QC, Canada, in 2013.

In October 2013, he started his Postdoctoral Research with the Department of Geography and Remote Sensing, McGill University, where he has worked on multispectral image processing and pattern recognition for cultural heritage preservation, forensic applications, and remote sensing. His research was supported by FRQNT. Since September 2017, he has been with the Department of Computer Science, Sultan Qaboos University, Muscat, Oman, as an Assistant Professor.

Dr. Hedjam is a member of IEEE Geoscience and Remote Sensing Society.

Abdelhamid Abdesselam received the B.Sc. degree in computer science from Universite Houari Boumediene d'Alger, Bab Ezzouar, Algeria, in 1984, the M.Sc. degree in pattern recognition and artificial intelligence from Universite Paul Sabatier de Toulouse, Toulouse, France, in 1985, and the Ph.D. degree in computer science from Institut National Polytechnique de Toulouse, Toulouse, France, in 1991.

He was an Assistant Professor with the Faculty of Information Technology, Universiti Malaysia Sarawak, where he headed the Imaging and Spatial Information Systems Research Group and participated in setting and developing the Masters in Spatial Technology Program. He is currently an Associate Professor with the Department of Computer Science, Sultan Qaboos University, Muscat, Oman. His research interests include image processing, computer vision, pattern recognition, and machine learning.

Farid Melgani received the State Engineer degree in electronics from the University of Batna, Algeria, in 1994, the M.Sc. degree in electrical engineering from the University of Baghdad, Iraq, in 1999, and the Ph.D. degree in electronic and computer engineering from the University of Genoa, Italy, in 2003.

He is a Full Professor of telecommunications at the Department of Information Engineering and Computer Science, University of Trento, Italy, where he teaches pattern recognition, machine learning, and digital transmission. He is the Head of the Signal Processing and Recognition Laboratory, and the Coordinator of the Doctoral School in Industrial Innovation at the same university. His research interests are in the areas of remote sensing, signal/image processing, pattern recognition, machine learning and computer vision. He is coauthor of more than 220 scientific publications. He is currently an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, International Journal of Remote Sensing, and IEEE Journal on Miniaturization for Air and Space Systems.